UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/994,023	11/27/2001	Andrew Mark Nightingale	550-286	1196

23117 7590 06/12/2009

NIXON & VANDERHYE, PC
901 NORTH GLEBE ROAD, 11TH FLOOR
ARLINGTON, VA 22203

EXAMINER	
ALHIJA, SAIF A	

ART UNIT	PAPER NUMBER
2128	

MAIL DATE	DELIVERY MODE
06/12/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

RECORD OF ORAL HEARING

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte ANDREW MARK NIGHTINGALE

Appeal 2009-002102
Application 09/994,023
Technology Center 2100

Oral Hearing Held: May 12, 2009

Before LEE E. BARRETT, JEAN R. HOMERE,
and JOHN A. JEFFERY, Administrative Patent Judges

ON BEHALF OF THE APPELLANT:

JOHN R. LASTOVA, ESQUIRE
NIXON & VANDERHYE, PC
901 NORTH GLEBE ROAD, 11TH FLOOR
ARLINGTON VA 22203

The above-entitled matter came on for hearing on Tuesday, May 12, 2009, commencing at 9:00 a.m., at The U.S. Patent and Trademark Office, 600 Dulany Street, Alexandria, Virginia, before Jack L. Becker, Notary Public.

MS. BOBO-ALLEN: Calendar No. 5, Appeal No. 2009-2102, Mr. Lastova.

JUDGE BARRETT: Okay, thank you. Good morning.

MR. LASTOVA: Good morning. Today's appeal on the technology relates to efficient testing and validation of integrated circuit designs. The problem the inventor in this case was trying to solve related to the testing of system-on-chip, SOC, designs which integrate a large number of functional elements on to a single integrated circuit. That requires a lot of testing and validation before the designs can be reliably released for manufacture, that's the bottleneck, all right. That's the problem we're trying to solve and the inventor does solve that.

I'm going to just briefly explain how the inventor solved that using Figure 1 in the drawings. And if you take a look there real quickly, you'll see that there are signal interface controllers 8 and 9 right in the middle of the page there. They're connected to hardware simulation 4, and they perform simulation by applying stimulus signals and re-response signals from the simulation 4, all right. Then up at the top of Figure 1, you have the test scenario manager 10. It sends these signal interface controllers test scenario-controlling messages that specify the simulation actions that we want performed and when those actions are to be performed. So, a significant point here is that these signal interface controllers 6 and 8 right in the middle of the page can be reused across multiple test designs. So that's quite a nice little feature we have there. The test manager simply says okay, here's these simulations I want performed in a rather high-level fashion, and it relies on these reusable signal interface controllers to transform these high-

level simulation actions into the detailed stimulus and response signals that we need to do the actual testing, all right.

Then you see the last element, there is a time generator 12. It generates some time-defining events for the manager 10 and the signal interface controllers 8 and 6 corresponding to the simulated time now, okay, not the real time, the simulated time used in the hardware simulator 4, all right.

Last point, you'll see in the manager 10 at the top, there a little box called scoreboard. That's the shared data memory in the claim that we'll talk about in a moment. And into that shared data memory, the signal interface controllers 6 and 8 here, just representative, of course, they can store data and read data independently of the simulated time. Okay, remember we talked about the time generator, the simulated test time, all right, and they can do this independently of that, and that's a significant point, all right. We'll get to reasons why that's significant in just a moment.

Let's get to the rejection. The -- all the claims now are rejected for anticipation, all right, based on the Rajsuman reference. And just briefly, Rajsuman teaches a software application being broken up into multiple subtasks, are scheduled and assigned to different verification units, VU, I'm going to use that acronym to be -- use a shorthand. If you look at Figure 5, that's the main figure for that case and that's the figure the Examiner seems to rely on the most. Figure 5 shows these VUs in sort of the center of the page 66. And it also shows on the right-hand side an emulator subsystem labeled DVS-6 that includes an arbitration unit 76 and a synchronization clock unit 75. And then you've also got a main system CPU at the bottom of

the page that's 62 that controls these control CPUs 671 through 676 inside each VU -- or attached to each VU over a bus 64.

Let's look at the Examiner's claim mapping, all right. So, for the claimed hardware simulator, the Examiner maps that on to silicon integrated circuits and he cites column 5, lines 41 to 48 for that. For the claimed system interface controllers, he maps that on to the verification units, or the VUs, 66.

JUDGE JEFFERY: Counsel, let me stop there and ask you a question about the verification units. Yes, the Examiner is mapping that to the claimed signal interface controllers. And it seems from the Answer the Examiner's making a big deal out of the fact that these verification units are tied together through a bus arrangement. And he seems to be suggesting that that functionality of the "interconnectedness," if you will, of the verification units that ultimately tie back to the main CPU at the bottom, which the Examiner says is the test scenario manager or corresponds to the test scenario manager, somehow that that provides a shared data memory as claimed. And as I take it, your position is that the fact that these are tied together by a bus doesn't in any way, shape, or form provide a shared data memory, let alone a shared data memory in the test scenario manager because that's where -- according to the claim, that's where the shared memory has to be.

MR. LASTOVA: That's right.

JUDGE JEFFERY: Okay. So, my question to you is in column 7 of the reference where it discusses the verification units themselves, they, they each contain a CPU. I'm actually starting around line 60 of column 7.

MR. LASTOVA: Right.

JUDGE JEFFERY: And it talks about the verification units and that they have their own CPUs and they're tied together with buses.

MR. LASTOVA: That's right.

JUDGE JEFFERY: And it mentions that the CPUs of the verification units control data flow and so on and so forth, but it also mentions "response comparison" in line 63. Now it would suggest that the verification unit CPUs can somehow do some sort of analysis of each other's responses perhaps. It's hard to say from that.

MR. LASTOVA: I agree with you.

JUDGE JEFFERY: But in any event, even if that were done at the verification unit level, I, I am just -- would there be any kind of shared data memory to make that happen at the, the main CPU where it would be the -- ?

MR. LASTOVA: Right, I think I know where you're coming from this, and I think it's important that we try to keep clear in our minds what we're really looking at, right. When you look at Rajsuman, you see that they do have these CPUs and CPUs have memory, that's true. The question is, as you aptly point out, is does the, the claim as arranged -- is that taught here? And we specifically say that the test scenario manager, which, of course, is the main CPU 62, okay, that includes the shared data memory. Now, you were talking about -- so, that's not taught, that's clearly not taught, all right. You were suggesting there's some kind of connectionality between these various functions. Sure, they're connected by a bus, and you saw in our brief, I did show from an encyclopedia a definition of a bus, and a bus is a data path, and as we know, a data path is not the same thing as a memory. Now, it can hook up to a memory which is a device that's addressable and so forth, but I don't think there's much argument that a data path isn't memory.

So where I was coming from in my reply, as well as in my brief, and to address your point, is that this connection function really doesn't get to the heart of what we're claiming here. We're not talking about do they pass -- are there signals being passed between these various units. We're asking do the VUs, okay, do they say, okay -- and we'll get to the independent of the simulated time -- do they actually read and write to each other using a shared memory, okay, in the main CPU 62? And the answer is no, they don't do that, okay.

JUDGE JEFFERY: Even irrespective of the timing issue, I mean that, that seems to be missing from the reference even without the timing aspect.

MR. LASTOVA: That's right, that's right.

JUDGE JEFFERY: Simulated time – "being independent of simulated time." I mean even, even without that, it's -- I mean the threshold question here is: Do you have a shared data memory in the main CPU?

MR. LASTOVA: And I think the answer to the threshold question is no, and we literally could stop there for anticipation, all right. What I was trying to also point out is that some of the strained -- you used the word "somehow" -- strained manipulations in the rejection use these sort of connections, data path, data bus, and so forth. I was trying to point out it's not even clear that there's even a shared memory. I'm not sure that that's even truly taught, explicitly taught. It's certainly not taught in the main CPU, it's certainly not taught with the, with the functionality that we've claimed that the signal interface controllers can read to it and write to it. So, we've got that.

Let's go to another threshold issue that I think will just sort of "seal the deal," if you will. And that relates to this issue that I was alluding to

with respect to the timing. And I think there's a little confusion by the Eexaminer when the Examiner is looking at this timing, and he's just using the word "time," okay. He's not making the distinction between simulated time and real time. He doesn't understand or, or at least is ignoring for purposes of the rejection, the fact that simulated time, and you recall from Figure 1 it's easier to see there, the time generator generates some simulated time to be used in the hardware simulation, okay. That's the testing, all right. Remember now, the mapping that the Examiner provided was the simulations are the ICs being tested, and that makes sense, sure. The things we're going to be testing are the things we're going to revolve around the simulated time. What ends up happening, though, is when he does the rejection, he steps away from the simulation and starts looking at the synchronization unit 75. Well, the synchronization unit 75 is synchronizing the VUs, okay, not the ICs. That, that's a real-time functionality there, that's real-time transfers over the bus and so forth. So what's happening here is that there isn't a reading and writing to this shared data memory inside the main CPU 62, all right, independently -- I'm reading from the quote -- quoted from the claim now – "independently of advancement of the simulated time by the messages specifying time-defined events, said data being readable from the shared memory by another signal interface controller." And the significance here is, of course, is an advantage in our system is that because we have this distinction, all right, this independence between simulated time and real time, we can actually pass data over a data bus, for example, all right, and that doesn't require simulated time. Simulated time can actually stop, the real time continues on, and that's

something that that distinction and that flexibility simply isn't taught in Rajsuman.

So what we really have here is we have actually multiple grounds to reverse the anticipation rejection. We have a shared memory that I think is missing because it doesn't -- we don't teach a shared memory that's readable and writable by the signal interface controllers. We also don't have the shared memory actually in the test scenario manager as claimed. And we also don't have the storing of data in that shared memory independent of advancement of simulated time. So there are three different "holes," if you will, in the rejection of the independent claims, and those are the three features any one of which I think require reversal of the Examiner's rejection. So, actually you've, you've moved me through, Judge Jeffery, my, my little prepared speech here quickly, so I have a few additional moments here if there are questions, but if not, that's my presentation.

JUDGE BARRETT: Questions? That's it, thank you.

MR. LASTOVA: Thank you.

JUDGE BARRETT: The court reporter might have some, I don't know.

COURT REPORTER: What's the spelling of Rajsuman?

MR. LASTOVA: Yes, right here, yes, okay. So it's R A J S U M A N. And it was VU like V as in Victor, V. Thank you very much, have a good day.

JUDGE BARRETT: Thank you.

(Whereupon, the hearing concluded at 9:39 a.m. on May 12, 2009.)